

# 113 年特種考試地方政府公務人員及離島地區公務人員考試試題

考試別：地方政府公務人員、離島地區公務人員考試

等 別：三等考試

類 科：資訊處理

科 目：資料結構

曹勝 老師

一、考慮下面以虛擬碼 (Pseudocode) 表示的遞迴演算法，請回答相關問題：

```
Algorithm  $Q(n)$ 
  if  $n=1$ 
    return 1
  else return  $Q(n-1)+2\times n-1$ 
```

(一)列出虛擬碼中  $Q(n)$  的遞迴關係式，並說明此虛擬碼最終計算的是什麼？(5 分)

(二)用遞迴函式表示此虛擬碼所使用的乘法運算次數，並用漸進式符號 Big-O 表示此遞迴函式的成長速率。(5 分)

(三)以遞迴函式表示此虛擬碼的執行時間  $T(n)$  並說明其時間複雜度 (以 Big-O 表示)。(10 分)

1.《考題難易》★★★★

2.《破題關鍵》：遞迴關係式  $Q(n)$  描述奇數序列總和，最終計算結果為  $n$  的平方。遞迴執行時間  $T(n)$  線性增長，時間複雜度為  $O(n)$ 。

【擬答】

(一)遞迴關係式與虛擬碼的功能

根據虛擬碼的描述， $Q(n)$  的遞迴關係式為：

$$Q(n)=Q(n-1)+(2n-1)$$

邊界條件為：

$$Q(1)=1$$

功能分析

每次遞迴的新增項是  $2n-1$ ，即奇數序列的第  $n$  項。

將  $Q(n)$  展開： $Q(n)=Q(n-1)+(2n-1)=Q(n-2)+(2(n-1)-1)+(2n-1)=\dots$  當  $n=1$  時停止，因此： $Q(n)=1+3+5+\dots+(2n-1)$

根據數列求和公式，這是一個連續奇數序列的總和，其結果為： $Q(n)=n^2$

此虛擬碼最終計算的是  $n^2$ 。

(二)遞迴函式中的乘法運算次數與 Big-O 表示

在虛擬碼中，沒有明顯的乘法運算符號。所有數值計算都是加法或減法，因此沒有顯式的乘法運算。Big-O 表示因為沒有進行乘法運算，所以乘法次數是 0，漸進成長速率為：

$$O(1)$$

(三)執行時間  $T(n)$  的遞迴函式與時間複雜度

在虛擬碼中，每次執行  $Q(n)$  時，遞迴調用一次  $Q(n-1)$  並進行一次加法操作。

因此，執行時間可表示為： $T(n)=T(n-1)+O(1)$

邊界條件為： $T(1)=O(1)$

## 公職王歷屆試題 (113 地方特考)

展開遞迴式： $T(n)=T(n-1)+O(1)=T(n-2)+O(1)+O(1)=\dots=O(1)\times n$

時間複雜度：

最終的遞迴展開結果顯示，執行時間與  $n$  呈線性關係。使用 Big-O 表示，時間複雜度為： $O(n)$

### 二、請回答下列關於二元樹 (Binary Tree) 的問題：

(一)一個算術運算式 (Arithmetic Expression) 可以用一個二元樹表示，稱為算術運算樹 (Expression Tree)，請將下列算術運算式以算術運算樹表示。(5 分)

$$(((5+1)\times 3 - (7+2))/((2\times 8)+5)/7)$$

(二)請判斷下列敘述是否正確：(5 分)

“一個算術運算樹是一個滿二元樹 (Full Binary Tree, or Proper Binary Tree)”

(三)子題(一)中的算術運算式是何種順序的運算表示式？請利用其算術運算樹將此運算式表示為一前序表示式 (Preorder Expression)，並說明其過程。(5 分)

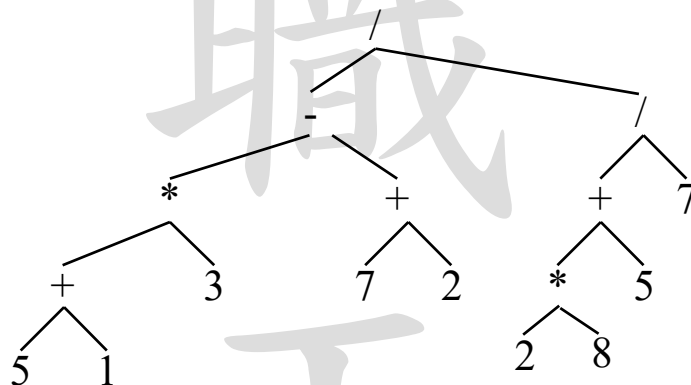
(四)請敘述如何以子題(一)的算術運算樹計算出算術運算式的值，並逐步表示其過程。(10 分)

1. 《考題難易》★

2. 《破題關鍵》：算術運算式可透過樹狀結構表示，運算符作為節點，數值作為葉子。採用後序遍歷自底向上計算算術運算樹的值。

#### 【擬答】

(一)算術運算式的算術運算樹表示



(二)在算術運算樹中，所有的非葉節點都必須是二元運算符(例如 +, -, \*, /)，而每個二元運算符都必須有兩個子節點(左操作數和右操作數)。因此，算術運算樹滿足滿二元樹的定義：每個非葉節點都有兩個子節點。因此敘述：「一個算術運算樹是一個滿二元樹(Full Binary Tree, or Proper Binary Tree)」是正確的。

(三)前序表示式(Preorder Expression)按以下順序生成：

訪問根節點

遞迴處理左子樹

遞迴處理右子樹

對應子題 (一)的算術運算樹：

根節點為 /。

遞迴生成左子樹的前序表示式： $- * + 5 1 3 + 7 2$

遞迴生成右子樹的前序表示式： $/ + * 2 8 5 7$

合併結果： $/ - * + 5 1 3 + 7 2 / + * 2 8 5 7$

(四)計算算術運算式的值

1. 自底向上(Bottom-Up)：先計算葉節點，再逐層向上計算。

2. 使用後序遍歷(Postorder Traversal)：按照「左子樹 → 右子樹 → 根節點」的順序訪問並計算值。

3. 逐步計算過程：

左子樹：

計算  $(5 + 1) = 6$

計算  $6 * 3 = 18$

計算  $(7 + 2) = 9$

計算  $18 - 9 = 9$

右子樹：

計算  $(2 * 8) = 16$

計算  $16 + 5 = 21$

計算  $21 / 7 = 3$

根節點：計算  $9 / 3 = 3$

最終結果：算術運算式的值為 3

三、在許多應用中，往往需要以物件的優先權來進行處理，為了區別物件的優先順序，我們可以簡單地賦予物件一個鍵值 (Key) 來代表優先權，此鍵值通常是一個數值可以用來區別物件前後順序。在此，我們考慮物件的鍵值是一個數值而其值愈小，物件的優先權愈高，優先佇列 (Priority Queue) 則是一種以物件的優先權來管理物件的資料結構。

(一) 請說明優先佇列的抽象資料型態 (abstract data type, ADT) 定義。(10 分)

(二) 給定一個最小二元堆積 (Minimum Heap)  $H$  與一個鍵值  $k$ ，在  $H$  中快速地找出所有鍵值小於或等於  $k$  的資料物件。請描述一個有效的方法，此方法所花的時間 (或運算量) 與欲找出的資料物件之數量成線性比例。(5 分)

1. 《考題難易》★★★

2. 《破題關鍵》：優先佇列是以優先權管理資料的抽象資料型態，支援插入、刪除最優物件等操作。利用最小堆積特性，遞迴篩選鍵值  $\leq k$  的節點，達到高效搜尋。

【擬答】

(一) 優先佇列的抽象資料型態 (ADT) 定義

優先佇列是一種以優先權來管理資料的抽象資料型態，常用於需要依據優先順序處理物件的應用 (例如任務排程、事件處理)。其主要特性如下：

1. 資料操作

(1) 插入 (Insert / Enqueue)：將一個物件加入佇列。

(2) 刪除並返回最優先物件 (Extract-Min / Dequeue)：移除並返回佇列中優先權最高的物件 (在最小優先佇列中，鍵值最小的物件具有最高優先權)。

(3) 查看最優先物件 (Peek / Find-Min)：不刪除的情況下，返回優先權最高的物件。

2. 性質

每個物件都有一個鍵值 (Key) 代表其優先權，鍵值越小，優先權越高 (最小優先佇列)。優先佇列支持動態操作，即插入新元素和刪除最優先物件的時間複雜度通常較低 (如  $O(\log n)$ )。

(二) 在最小堆積 (Minimum Heap) 中快速找出所有鍵值  $\leq k \leq k$  的資料物件

在最小堆積中，每個節點的鍵值小於或等於其所有子節點的鍵值。這一特性可以用來高效篩選所有鍵值  $\leq k \leq k$  的物件。演算法如下：

1. 初始化：

建立一個空的結果集合 result，從堆積的根節點開始進行搜尋。

2. 遞迴(或使用堆疊模擬的迴圈)過程：

若當前節點的鍵值  $k \leq k$ ，將其加入 result 集合。

遞迴檢查當前節點的子節點。若當前節點的鍵值  $> k$ ，則剪枝(Prune)操作：停止處理該節點及其子節點，因為其子節點的鍵值必然大於  $k$ 。

3. 結束條件：

當所有符合條件的節點都已處理完時，返回結果集合 result。

4. 演算法實現

以遞迴方式的虛擬碼表示如下：

```
function FindKeys(heap, index, k, result):
```

```
    if index >= heap.size:
```

```
        return
```

```
    key = heap[index]
```

```
    if key <= k:
```

```
        result.add(heap[index]) # 加入結果集合
```

```
        FindKeys(heap, left_child(index), k, result) # 遞迴檢查左子節點
```

```
        FindKeys(heap, right_child(index), k, result) # 遞迴檢查右子節點
```

```
    else:
```

```
        return # 剪枝：若當前節點鍵值大於 k，則不再繼續搜索
```

5. 時間複雜度分析

堆積的特性確保只有可能包含鍵值  $\leq k$  的節點會被訪問，其他節點會被剪枝，避免不必要的檢查。若結果中有  $m$  個物件，則演算法的時間複雜度為  $O(m)$ ，即與找出的物件數量成線性比例。



# 資訊處理榮耀上榜

110地特四等 <b>台北市狀元</b>	110地特四等 <b>金門縣狀元</b>	111普考 <b>全國榜眼</b>	111地特三等 <b>金門縣榜眼</b>	110地特三等 <b>桃園市第四</b>	110地特三等 <b>花東區第四</b>	111地特四等 <b>台北市第八</b>	110普考 <b>全國第十</b>
于○	吳○展	羅○昌	李○杰	丁○妮	羅○哲	吳○進	陳○廷

高 考   孫○宇	高 考   于 ○	高 考   廖○湖	高 考   郭○楷	普 考   湯○安	普 考   王○如	普 考   陳○明	普 考   徐○翔
高 考   邱○銘	高 考   王○禎	高 考   黃○穎	高 考   廖○仲	普 考   林○慧	普 考   邱○志	普 考   鄭○然	普 考   楊○億
高 考   高○銘	高 考   施○晟	高 考   賴○全	高 考   羅○昌	普 考   方○天	普 考   許○毅	普 考   吳○翰	普 考   林○廷
高 考   林○慧	高 考   方○天	高 考   黃○迪	高 考   劉○廷	普 考   高○茗	普 考   鄧○泓	普 考   曾○瑄	普 考   許○文
高 考   傅○培	高 考   程○瑜	高 考   張○偉	高 考   李○庭	普 考   鄭○豪	普 考   宋○嶸	普 考   賴○全	普 考   楊○翔
高 考   梁○秀	高 考   王○如	高 考   郭○哲	高 考   曾○瑄	普 考   林○挺	普 考   黃○迪	普 考   張○慧	普 考   林○勳
高 考   施○宇	高 考   楊○滂	高 考   胡○紘	高 考   于 ○	普 考   郭○蕭	普 考   劉○廷	普 考   劉○銘	普 考   詹○宇
高 考   劉○瑜	高 考   傅○華	高 考   許○傑	高 考   陳○宇	普 考   黃○倫	普 考   張○偉	普 考   陳○堂	普 考   于○
高 考   鄧○泓	高 考   郭○蕭	高 考   陳○廷	普 考   王○文	普 考   盧○銘	普 考   褚○華	普 考   廖○仲	普 考   邱○智
高 考   涂○璋	高 考   林○廷	高 考   陳○明	普 考   梁○秀	普 考   朱○毅	普 考   李○庭	普 考   楊○雙	普 考   于○恩

普考榜眼
高普雙榜
半年考取
應屆考取

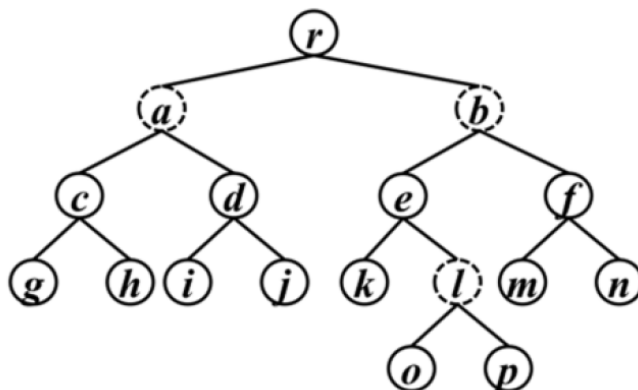
非常感謝補習班提供的題目資源，使得真正在上場考試時，總有這樣的心得：「好耶，這題我完全會寫。」當下真的非常開心，因為我先前沒有去報考其他考科練筆，完全依靠補習班資源，有這樣的結果實在太好了。

羅○昌 高普考-資訊處理

四、關於紅黑樹 (Red Black Tree) 與(2,4)-樹((2,4)-Tree)：

(一)請分別說明紅黑樹與(2,4)-樹的定義。(10 分)

(二)考慮下面的紅黑樹(實線節點代表黑色節點，虛線節點代表紅色節點)，代表節點的字元符號可視為鍵值，請說明如何將此紅黑樹轉換為一個(2,4)-樹，並將其結果畫出。此外，請申論轉換的(2,4)-樹是否唯一。(10 分)



(三)請說明為何一個有  $n$  個節點(鍵值)的紅黑樹其高度是  $O(\log n)$ 。(5 分)

1. 《考題難易》★★★
2. 《破題關鍵》：紅黑樹平衡性確保  $O(\log n)$  高度，紅節點與父節點合併形成 (2,4)-樹；紅黑樹與 (2,4)-樹平衡原理相同，均維持結構穩定性與搜尋效率。

【擬答】

(一)紅黑樹與 (2,4)-樹的定義

1. 紅黑樹 (Red Black Tree)

是一種平衡的二元搜尋樹，附帶以下性質：

節點為紅色或黑色。

根節點必須是黑色。

紅節點的子節點必須是黑色(即沒有兩個相鄰的紅節點)。

從任何節點到其所有葉節點的每條路徑都包含相同數量的黑色節點(稱為「黑色高度」)。

所有葉節點為空節點(Nil)，並視為黑色。

2. (2,4)-樹 ((2,4)-Tree)

是一種 B 樹 (B-Tree)的特例，符合以下特性：

每個節點最多包含 4 個子節點和 3 個鍵值。

所有的葉節點位於相同深度。

每個內部節點包含  $k$  個鍵值和  $k+1$  個子節點，且鍵值排列有序。

(2,4)-樹始終保持平衡。

(二)紅黑樹轉換為 (2,4)-樹

紅黑樹的紅色節點可視為將其與其父節點合併的狀態。具體而言：

如果紅色節點存在，將其鍵值視為父節點的擴展。

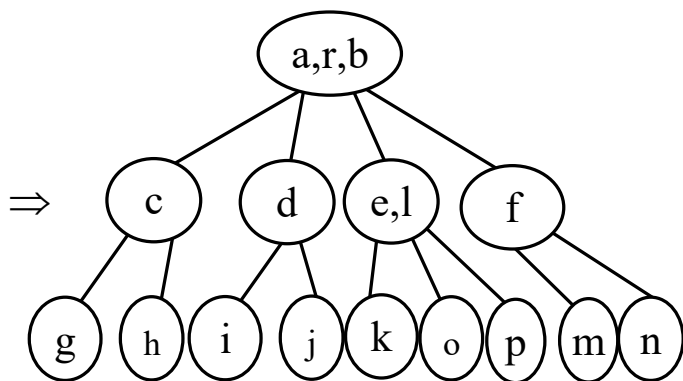
這種合併將產生多鍵值節點，對應 (2,4)-樹的結構。

根據圖中的紅黑樹：

將每個紅色節點與其父節點合併，產生相應的多鍵值節點。

例如，紅色節點  $a$  與  $r$  合併， $d$  與其父節點合併，依此類推。

結果圖示



此轉換的結果是唯一的，因為每個紅色節點都按照其父子關係進行固定合併，並維持鍵值的排序與結構平衡。

(三)紅黑樹高度為  $O(\log n)$ 的說明

紅黑樹是一種平衡二元搜尋樹，其平衡條件(例如紅節點限制和黑高一致性)確保其高度不會過高。樹的最大高度出現在所有路徑都交替紅色和黑色節點的情況。

紅黑樹每條根到葉節點的最短路徑(僅包含黑節點)至少是其最長路徑長度的一半。

在  $n$  個節點的情況下，樹的高度  $h$  滿足： $h \leq 2 \times \lceil \log_2(n+1) \rceil$

因此，紅黑樹的高度為  $O(\log n)$ 。

志光 學儒 保成

# 順利考取有訣竅

**一年考取 高普雙榜**  
112高普考資訊處理  
涂○瑋

各科老師都會詳細解說考試重點，讓我好好學習那些不熟悉的知識。考前總複習班各科老師以最短的時間來幫大家複習重點及預測考試的命題趨向，對於一整年的課程有畫龍點睛的功效。

**非本科系 高普雙榜**  
112高普考資訊處理  
傅○華

老師都是以實例還有時事去講解，所以非常的清楚也好記憶，跟著老師的步驟，配合每一次章節結束的歷屆試題講解，就知道遇到題目時該如何解題，讓非本科生的我受益良多。

想了解更多訣竅？  
歡迎至 志光.學儒.保成 全國門市洽詢

公職王歷屆試題 (113 地方特考)

五、下面的矩陣  $M$  是表示一個無向圖  $G=(V, E)$  的相鄰矩陣 (Adjacency Matrix)， $V$  與  $E$  分別為節點與邊的集合：

	$a$	$b$	$c$	$d$	$e$	$f$	$g$
$a$	0	1	0	1	1	1	0
$b$	1	0	1	0	1	1	0
$c$	0	1	0	0	0	1	1
$d$	1	0	0	0	0	0	1
$e$	1	1	0	0	0	0	0
$f$	1	1	1	0	0	0	1
$g$	0	0	1	1	0	1	0

(一)請畫出此無向圖  $G$ 。(10 分)

(二)若以字母順序為考量對  $G$  進行廣度優先搜尋 (Breadth-First Search, BFS)，因此將由節點  $a$  開始，請繪出尋訪完後所產生的 BF 樹 (Breadth-First (BF) Tree)。(5 分)

1. 《考題難易》★★
2. 《破題關鍵》：針對無向圖  $G$ ，根據相鄰矩陣畫出節點連接；廣度優先搜尋則依字母順序展開，確保層次明確並形成 BF 樹。

【擬答】

(一)畫出無向圖  $G$

根據相鄰矩陣，每個節點的連接關係如下：

節點  $a$ : 與  $b, d, e, f$  相連

節點  $b$ : 與  $a, c, e, f$  相連

節點  $c$ : 與  $b, f, g$  相連

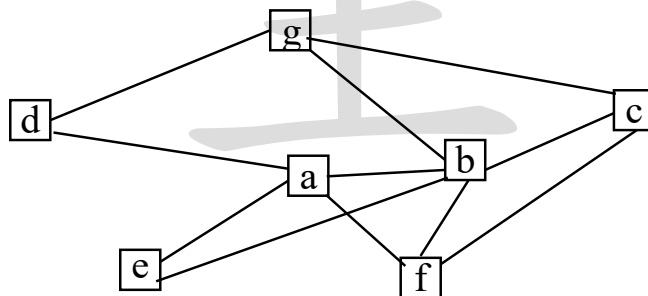
節點  $d$ : 與  $a, g$  相連

節點  $e$ : 與  $a, b$  相連

節點  $f$ : 與  $a, b, c, g$  相連

節點  $g$ : 與  $c, d, f$  相連

根據這些連接關係畫出無向圖  $G$ ：



(二)廣度優先搜尋 (BFS) 的 BF 樹

從節點  $a$  開始，按字母順序進行廣度優先搜尋，過程如下：

1. 初始節點:  $a$

可選子節點依字母順序:  $b, d, e, f$

訪問順序:  $b \rightarrow d \rightarrow e \rightarrow f$

2. 節點  $b$  的子節點:

$b$  已訪問，僅新增  $c$  為下一層節點。

公職王歷屆試題 (113 地方特考)

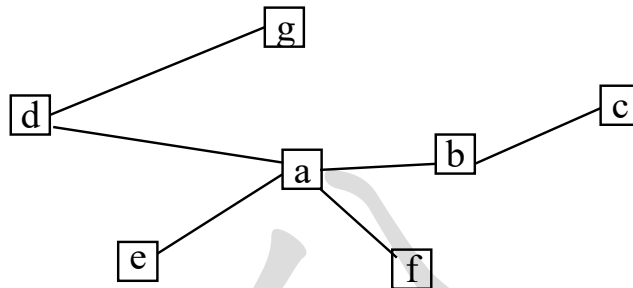
3. 節點 d 的子節點:

d 已訪問，僅新增 g 為下一層節點。

4. 節點 e, f, c, g 的子節點:

所有子節點已在之前訪問。

5. 最後形成的 BF 樹



志光 學儒 保成 高普考.地方特考

## 資訊處理上榜養成規劃

- 基礎架構課程**  
考科概念建立  
適應教學模式
- 正規課程**  
規劃完整堂數  
雙循環雙師資
- 進階課程**  
獨家  
圖解階段複習  
解題技巧灌輸
- 趨勢講座**  
時事考點補充  
命題趨勢分析
- 題庫班**  
精選題目教學  
學習快速解題
- 總複習班**  
科目重點整理  
考前強化記憶

詳細課程內容, 歡迎至志光學儒保成全國門市洽詢